* 接口

IComponent, ISpout, ISpout -> IRichSpout,IRichBolt, IBasicBolt

BaseComponent, BaseRichSpout(open,nextTuple), BaseRichBolt(prepare,execute, cleanup), BaseBasicBolt

IBatchBolt(execute, finishBatch, prepare), BaseBatchBolt

IStateSpout, IRichStateSpout

声明字段: declareOutputFields

uilder.setBolt().shuffleGrouping("word-reader");

IBatchSpout

ack，lose()，**emitBatch**，getComponentConfiguration，getOutputFields．open

FixedBatchSpout　implements IBatchSpout

persistentAggregate 存储更新状态

* 分组

shuffle group随机分组: 随机分

fields group按字段分组: 按字段分组

all grouping全复制分组: 发给所有的task

globle grouping 全局分组: 唯一的task(最小task ID)

none grouping不分组,随机,为将来预留.

direct grouping指向分组: 指向型数据流上使用,执行组件.

local or shffle grouping本地或随机分组: 随机类似.发送给同一个worker内的bolt task.

PartialKeyGrouping 部分关键字分组,考虑到下游bolt均衡情况.

* 锚定tuple: 建立输入tuple和衍生出的tuple间的对应关系.下游的tuple可以进行应答确认,超时或报错.
* 多task原因

灵活的通过rebalance进行不离线系统伸缩. task数量运行后不能变动

便于测试,大规模系统正确性.两个线程运行多个task.

实际运行1个 executor 1 task .

* Hook 钩子:在storm内部插入自定义的代码来运行任意数量的事件.
* Trident状态: 事务和不透明事务.非事务.

Trident具有功能：连接，聚合，分组，函数，过滤器

* **Trident 5类型操作:**本地分区: 重新分区: 聚合: 流分组(groupBy),合并与连接(Merges,join)

本地分区, 函数,过滤器,分区聚合CombinerAggregator, ReducerAggregator, Aggregator, 状态查询和分区持久化, 投影

一个**CombinerAggregator**仅输出一个tuple（该tuple也只有一个字段）。每收到一个输入tuple，CombinerAggregator就会执行**init**()方法（该方法返回一个初始值），并且用**combine**()方法汇总这些值，直到剩下一个值为止（聚合值）。如果partition中没有tuple，CombinerAggregator会发送**zero**()的返回值。下面是聚合器Count的实现：

* CombinerAggregation好处: 自动优化计算：先做局部聚合操作，然后再通过网络传输tuple进行全局聚合。

ReducerAggregator使用init()方法产生一个初始值，对于每个输入tuple，依次迭代这个初始值，最终产生一个单值输出tuple。下面示例了如何将Count定义为ReducerAggregator：

Aggregator可以输出任意数量的tuple，且这些tuple的字段也可以有多个。

* Trident有另外一种更新State的方法叫做persistentAggregate。